Europäisches Patentamt | European Patent Office | Office européen des brevets

Bescheinigung | Certificate | Attestation

Die angehefteten Unterlagen stimmen mit der ursprünglich eingereichten Fassung der auf dem nächsten Blatt bezeichneten europäischen Patentanmeldung überein.

The attached documents are exact copies of the European patent application described on the following page, as originally filed.

Les documents fixés à cette attestation sont conformes à la version initialement déposée de la demande de brevet européen spécifiée à la page suivante.

Patentanmeldung Nr. Patent application No. Demande de brevet n°

04090263.7

Der Präsident des Europäischen Patentamts;
Im Auftrag

For the President of the European Patent Office

Le Président de l'Office européen des brevets
p.o.

R C van Dijk

Anmeldung Nr:
Application no.: 04090263.7
Demande no:

Anmeldetag:
Date of filing: 29.06.04
Date de dépôt:

Anmelder/Applicant(s)/Demandeur(s):

FRAUNHOFER-GESELLSCHAFT ZUR FÖRDERUNG DER
ANGEWANDTEN FORSCHUNG E.V.
Leonrodstrasse 54
80636 München
ALLEMAGNE

Bezeichnung der Erfindung/Title of the invention/Titre de l'invention:
(Falls die Bezeichnung der Erfindung nicht angegeben ist, siehe Beschreibung.
If no title is shown please refer to the description.
Si aucun titre n'est indiqué se referer à la description.)

Method and apparatus for automatic online detection and classification of anomalous objects in data stream

In Anspruch genommene Priorität(en) / Priority(ies) claimed /Priorité(s) revendiquée(s)
Staat/Tag/Aktenzeichen/State/Date/File no./Pays/Date/Numéro de dépôt:

Internationale Patentklassifikation/International Patent Classification/
Classification internationale des brevets:

G06K9/00

Am Anmeldetag benannte Vertragstaaten/Contracting states designated at date of filing/Etats contractants désignées lors du dépôt:

AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HU IE IT LU MC NL
PL PT RO SE SI SK TR LI

Method and apparatus for automatic online detection and classification of anomalous objects in data stream

The invention relates to a method for automatic online detection and classification of anomalous objects in a data stream according to claim 1 and an system to that aim according to claim 14.

In practical applications data analysis it is often necessary to evaluate the content of datasets so that the contents belong to certain classes.

One example would be the classification of measurements into normal and anomalous classes. The mathematical boundary between "normal" and "anomalous" is usually a mathematical condition which is either satisfied or not satisfied.

The following formulations are quite imprecise, as to the delineation of our invention from the previous are. I think the following delineation would be more appropriate (please feel free to re-formulate):
1. From previous art
(both patents by Vapnik plus the following papers:
Cortes, C. and Vapnik, V. "Support Vector Networks". Machine Learning, 1995, 20:273-297
K.-R.~Müller and S. Mika and G. Rätsch and K.~Tsuda and B.~Schölkopf: "An Introduction to Kernel-Based Learning Algorithms", IEEE Transactions on Neural Networks, 2001, 12:181-201)
it is known how to create an adaptable classification boundary as a result of an offline (batch) training process. Here some terminology should be precisely set up. For example, the Hartman's patent also uses the term "online" but the meaning of this term is "periodic re-training of the system when new batches of data is available". In contrast, our idea can update the system upon the arrival of a single point which is impossible in Hartman's case and any other SVM

patents, since their methods do not have a valid solution for a single point; ours does.

2. From previous art (
Porras, P.~A. and Neumann, P.~G., "Emerald: event monitoring enabling responses to anomalous live disturbances", Proc. National Information Systems Security Conference, 1997, pp. 353-365, and
Warrender, C. and Forrest, S. and Perlmutter, B., "Detecting intrusions using system calls: alternative data methods", Proc. {IEEE} Symposium on Security and Privacy, 1999, pp. 133-145)
it is known how to detect outliers online (in our sense, i.e. one example at a time) when the notion of normality is fixed in advance as a model.

It is not known, however, how to detect outliers in the continuous stream of data and at the same time construct the representation of normality on the fly or to dynamically adjust the representation with the arrival of new data. This precisely the scope of the new invention.

From prior art it is known to evaluate datasets offline to classify the information in the dataset according to certain conditions. The conditions can be either predefined and fixed or can be adaptable during the classification process. The latter "self-learning" case is often required in situations where the exact formulation of the mathematical conditions defining the classification is not known à priori.

The problem in real time application is that offline analysis is often not feasible or desireable.

From prior art it is known to classify data streams according to certain predefined classification conditions. But those techniques are not applicable to cases where the classifications condition cannot be known à priori.

One example for such an application would be the detection of an attack by a hacker to a computer system through a computer network.

The "normal" characteristics are known but it cannot in beforehand be defined how an attack would be represented in a datastream.
Again, one cannot in general claim that normal characteristics are known; in this case one would use techniques that use such descriptions. I would suggest the following formulation:
Explicit description of normality of operation of a computer system is extremely difficult and time-consuming; therefore, previously known methods which rely on such description cannot be applied.

It is only be known in advance that a certain deviation from the normal situation will take places.

The current invention related to such situation in which datasets are analysed in real time without definite knowledge of the classification criteria to be used in the analysis.

In the following the invention is described by the way of example by

Fig. 1 depciting a flow-diagram of one embodiment of the invention;

Fig. 2 depicting a detailed flow-diagram for the construction and updated of the geometric representation of normality;

Fig. 3 depicting a schematic view of an embodiment of the

inventive system for the detection of anomalous objects in connection with a computer network;

Fig. 4A-4C depicting examples for the initialisation of an embodiment of the invention;

Fig. 5A-5G depicting examples for the further processing of an embodiment of the invention.

A system and method are disclosed for online detection and classification of anomalous objects in continuous data streams.

In Fig. 1 the data flow of one embodiment is depicted.

The overall scheme of an embodiment of the system and the method is depicted in Fig. 1. The input of the system is a data stream 1000 containing normal and anomalous objects pertaining to a particular application. In the following it is assumed that the data stream 1000 is incoming data of a computer network. The system according to the invention is used to detect anomalous objects in said data stream 1000 which could indicate a hacker attack.

The data stream 1000 are data packets in communication networks.

Alternatively the data stream 1000 can be entries in activity logs, measurements of physical characteristics of operating mechanical devices, measurements of parameters of chemical processes, measurements of biological activity, and others which will summarized below.

The central feature of the method and the system according to the invention is that it can deal with continuous data streams 1000 in an online fashion. The term "continous" in this context means that data sets are received regularly or

irregularly (e.g. random bursts) by the system and processed one at a time.

The term "online" in this context means that the system can start processing the incoming data immediately after deployment without the extensive setup and tuning phase. The tuning of the system is carried out automatically in the process of its operation. This contrasts with an offline mode in which the tuning phase involves extensive training (such as with the systems bases on neural networks and support vector machines)of manual interaction (such as with expert systems).

The system can alternatively operate in the offline mode, whereby the data obtained from the data stream 1000 are stored in the database 1100 before being using in the further processing stages. Such mode can employed in the situations when the volume of the incoming data exceeds the throughout of the processing system, and intermediate buffering in the database is required.

It is possible to operate the application in a mixed mode (e.g. in ces the data is strongly irregular), in which at least a part of the total data stream is a continously incoming datastream 1000.

In this case, the system reads the data from the data stream 1000 as long is new data is available. If no new data is available, the system switches its input to the database and processes the previously buffered data. On the other hand, if the arrival rate of the data in the data stream 1000 exceeds the processing capacity of the system, the data is veered off into the database for processing at a later time. In this way, optimal utilization of computing resources is achieved.

Each of the incoming objects is supplied to a feature

extraction unit 1200, which performs the pre-processing required to obtain the features 1300 relevant for a particular application.

The purpose of the feature extraction unit is to compute, based on the content of the data, the set of properties ("features") suitable for subsequent analysis in an online anomaly detection engine 2000. These properties must meet the following requirements:

either

a) each property is a numeric quantity (real or complex), or

b) the set of properties forms a vector in an inner product space (i.e. computer programs are provided which take the said set of properties as arguments and perform the operations of addition, multiplication with a constant and scalar product pertaining to the said sets of properties), or

c) a non-linear mapping is provided transforming the sets of properties in the so-called Reproducing Kernel Hilbert Space (RKHS). The latter requirement can be satisfied by providing a computer program which takes the said sets of properties as arguments and computes a kernel function between the two sets of properties. The function realized by this program must meet (exactly or approximately) the conditions known as "Mercer conditions".

In the exemplary embodiment of the system, the features can be (but are not limited to)

- IP source address
- IP destination address
- TCP source port
- TCP destination port
- TCP sequence number
- TCP acknowledgement number

- TCP URG flag
- TCP ACK flag
- TCP PSH flag
- TCP RST flag
- TCP SYN flag
- TCP FIN flag
- TCP TTL field
- start of the TCP connection
- duration of the TCP connection
- number of bytes transmitted from the source to the destination
- number of bytes transmitted from the destination to the source

If the entire set of properties does not satisfy the imposed requirements as a whole, it can be split into subsets of properties. In this case, the subsets are processed by separate online anomaly detection engines.

Similarly to the data, the features can be buffered in the feature database 1400, if for some reason intermediate storage of features is desired.

Alternatively, if the incoming objects are such that they can be directly used in a detection/classification method, no feature extraction unit 1200 is necessary.

The features 1300 are then passed on to the online anomaly detection engine 2000.

The main step 2100 of the online anomaly detection engine 2000 comprises a construction and an update of an geometric representation of the notion of normality.

The online anomaly detection 2000 constitutes the core of the invention. The main principle of its operation lies in the construction and maintaining of a geometric representation of

normality 2200. The geometric representation is constructed in the form of a hypersurface (i.e. a manifold in a high-dimensional space) which depends on selected examples contained in the data stream and on parameters which control the shape of the hypersurface. The examples of such hypersurfaces can be (but are not limited to):

- a hyperplane
- a hypersphere
- a hyperellipsoid.

The online anomaly detection engine consists of the following components:
- the unit for construction and update of the geometric representation 2100
- the storage for the geometric representation 2200 produced by the unit 2100, and
- the anomaly detection unit 2300.

The output of an online anomaly detection engine 2000 is an anomaly warning 3100 which can be used in the graphical user interface, in the anomaly logging utilities or in the component for automatic reaction to an anomaly. In the exemplary embodiment for identification of hacker attacks, the consumers of an anomaly warning are, respectively, the security monitoring systems, security auditing software, or network configuration software.

Alternatively, the output of an online anomaly detection engine can be used for futher classification of anomalies. Such classification is carried out by the classification unit 4000 which can utilize any known classification method, e.g. a neural network, a Support Vector Machine, a Fischer Discriminant Classifier etc. The anomaly classification message 4100 can be used in the same security management components as the anomaly warning.

The geometric representation of normality 2200 is a parametric hypersurface enclosing the smallest volume among all possible surfaces consistent with the pre-defined fraction of the anomalous objects (see example in Fig. 4 and 5).

Said hypersurface is constructed in the feature space induced by a suitably defined similarity function between the data objects ("kernel function") satisfying the conditions under which the said function acts as an inner product in the said feature space ("Mercer conditions"). The update of the said geometric representation of normality 2200 involves the adjustment so as to incorporate the latest objects from the incoming data stream 1000 and the adjustment so as to remove the least relevant object so as to retain the encapsulation of the smallest volume enclosed by the geometric representation of normality 2200, i.e. the hypersurface. This involves a minimisation problem which is automatically solved by the system.

The construction and the update of the geometric representation of normality 2200 will be described in greater detail in connection with Fig. 2.

Once the geometric representation of normality 2200 is automatically updated, an anomaly detection 2300 is automatically performed by the online anomaly detection engine 2000 assigning to the object the

- status of a normal object, if the object falls into the volume encompassed by the geometric representation of normality 2200, or

- the status of an anomalous object, if the entry lies outside of the volume encompassed by the geometric representation of normality 2200.

The output of the online anomaly detection engine 2000 is used to issue the anomaly warning 3100 and/or to trigger the classification component 4000 which can utilize any known classification method such as decision trees, neural networks, support vector machines (SVM), Fischer discriminant etc.

The use of support vector machines in connection with the invention is described below in Appendix A.

The crucial difference is that - although we do employ almost exactly the machinery what is known by incremental SVM - we use it in a completely different regime - online learning - what cannot be handled by SVM unless the labels are also provided online. The only reference I would consider reasonable is that the output of our online anomaly detection engine can be supplied to the SVM (deployed in classification - not learning - mode, which does not require the label information) as a sort of trigger. This is the essence of the "new story" behind Fig. 1.

The geometric representation of normality 2200 can also be supplied to the classification component if this is required by the method.

In an exemplary embodiment of the construction and update of the geometric representation of normality 2100 the hypersurface representing the class of normal events is represented by the set of parameters $x_1, \ldots, x_n$ (i=1...n), one parameter for each object in the working set.

The size n of the working set is chosen in advance by the user There may be two reasons for this:

1. The data set is extremely large (tens of thousands examples), and maintaining all points in the equilibrium is computationally infeasible (too much memory is needed, or it takes too long). In this case, only the examples deemed most

relevant should be kept around. The weights of examples are related to the relevance of examples for classification; therefore, the weights are used in the relevance unit to determine the examples to be excluded.

2. The data has temporal structure, and we believe that only the newest elements are relevant. In this case we should through out the oldest examples; this is what the relevance unit does if temporal structure is indicated.

The parameters are further restricted to be non-negative, and to have values less than or equal to $C = 1/(n\nu)$, where ν is the expected fraction of the anomalous events in the data stream (e.g. 0,25 for 25% expected outliers), to be set by the user. This estimate is the only a à priori knowledge to be provided to the system. There may be some other, kernel-dependent parameters in the system. These parameters reflect some prior knowledge (if available) about the geometry of objects.

This is a very weak limitation since such estimates are readily available.

The working set it partitioned into the

"set 0" of the objects whose parameters x_k are equal to zero,

"set E" of the object whose parameters x_k are equal to C, and the

"set S" of the remaining objects.

The operation of the construction and update of the geometric representation of normality 2100 is illustrated in Fig. 2.

Upon the arrival of the data object k, the following three main actions are performed within a loop:

In step A2.5 the data entry is "imported" into the working set.

In step A2.6 the least relevant data object l is sought in the working set.

And in step A2.7 the data entry l is removed from the working set.

The importation and removal operations maintain the minimal volume enclosed by the hypersurface and consistent to the pre-defined expected fraction of anomalous objects.

For more complicated geometries a "volume estimate" might be an appropriate tem, since for more complicated surfaces such as the hyperellipsoid, the exact knowledge of a volume may not be available.

These operations are explained in more detail below. The relevance of the data object can be judged either by the time stamp on the object or by the value of parameter x_i assigned to the object.

See also the decription of the relevance unit (see Figure description).

The steps A2.1 to A2.4 are the initialization operations to be performed when not enough data objects have been observed in order to bring the system into equilibrium (i.e. not enough data to construct a hypersurface).

Construction of the hypersurface 2200 enclosing the smallest volume and consistent with the pre-defined expected fraction of anomalous objects amounts, as shown in the article

"Support Vector Data Description" by D.M.J. Tax and R.P.W. Duin, Pattern Recognition Letters, vol. 20, pages 1191-1.199, (1999), to solving the following mathematical programming problem:

$$\max_{\mu} \min_{\substack{=\le x\le C \\ a^T x+b=0}} W=-c^T x+\frac{1}{2}x^T Kx+\mu\ (a^T x+b\ ), \qquad (1)$$

where:

K is a n x n matrix that consists of evaluations of the given kernel function for all data points in the working set: $K_{i,j}$ = kernel(p_i , p_j).

For example, of the objects are vectors in the n-dimentional space, and the solution is sought in the linear feature space, the kernel function is evaluated as follows:

kernel(p_i, p_j) = \sum_{k=1}^n p_i^k p_j^k

As another example, if the solution is space in the features space of radial basis functions (which is n infinite-dimensional space, the kernel function is computed as:

kernel(p_i,p_j) = exp(-||p_i - p_j||/(2*gamma),

where gamma is the kernel parameter.

c is the vector of the numbers at the main diagonal of K,

a is the vector of n ones and b = -1.

The parameter C is related to the expected fraction of the anomalous objects.

The necessary and sufficient condition for the optimality of the representation attained by the solution to problem (1) is given by the well-known Karush-Kuhn-Tucker conditions.

When all the points in the working set satisfy the said conditions, the working set is said to be in equilibrium.

Importation of a new data objects into, or removal of an existing data object from a working set may result in the violation of the said conditions. In such case, adjustments of the parameters $x_1, \ldots, x_n$ are necessary, in order to bring the working set back into equilibrium.

An algorithm for performing such adjustments, based on the Karush-Kuhn-Tucker conditions, for a different mathematical programming problem - Support Vector Learning - was presented in the article "Incremental and Decremental Support Vector Learning" by G. Cauwenberghs and T. Poggio, Advances *in Neural* Information *Processing Systems 13, pages 409-415, (2001)*. By deriving the Karush-Kuhn-Tucker conditions for the problem (1), the necessary ingredients for the application of the method of Cauwenberghs and Poggio can be obtained.

The descriptions of the algorithms for importation and removal units are given in the description of the figures. They are somewhat different from the ones described by Cauwenberghs and Poggio, with more accurate processing of special cases crucial for the online learning.

Special care needs to be taken at the initial phase of the operation of the online anomaly detection engine as described in Fig. 2. When the number of data objects in the working set is less than or equal to $\lfloor \frac{1}{C} \rfloor$ (the greatest integer smaller than or equal to 1/C), equilibrium cannot be reached and the importation method cannot be applied.

The initialization steps A2.1 to A2.4 of the invention are designed to handle this special case and to bring the working set into the equilibrium after the smallest possible number of data objects has been seen.
See the explanation to the figure on the initialization unit of the plane sphere agent.

The exemplary embodiment of the online anomaly detection method in the system for detection and classification of computer intrusions is depicted in Fig. 3.

The online anomaly detection engine 2000 is used to analyse a data stream 1000 (audit stream) containing network packets and records in the audit logs of computers. The packets and records are the objects to be analysed.

The audit stream 1000 is input into the feature extraction component 1200 comprising a set of filters to extract the relevant features.

The extracted features are read by the online anomaly detection engine 2000 which identifies anomalous objects (packets or log entries) and issues an event warning if the event is discovered to be anomalous. Classification of the detected anomalous events is performed by the classification component 4000 previously trained to classify the anomalous events collected and stored in the event database.

The online anomaly detection engine comprises a processing unit having memory for storing the incoming data, the limited working set, and the geometric representation of the normal (non-anomalous) data objects by means of a parametric hypersurface; stored programs including the programs for processing of incoming data; and a processor controlled by the stored programs. The processor includes the components for construction and update of the geometric representation of normal data objects, and for the detection

of anomalous objects based on the stored representation of normal data objects.

The component for construction and update of the geometric representation receives data objects and imports it into the representation such that the smallest volume enclosed by the hypersurface and consistent with the pre-defined expected fraction of anomalous objects is maintained; the component further identifies the least relevant entry in the working set and removes it while maintaining the smallest volume enclosed by the hypersurface. Detection of the anomalous objects is performed by checking if the objects fall within or outside of the hypersurface representing the normality.

As an embodiment of the invention, the architecture of the system for detection and classification of computer intrusions is disclosed. The system consists of the feature extraction component receiving data from the audit stream; of the online anomaly detection engine; and of the classification component, produced by the event learning engine trained on the database of appropriate events.

In Fig. 4 and 5 the construction of the geometrical representation of normality 2200 is described, especially in conncection with the initialisation.

In order to find the optimal geometric representation of normality 2200 of a dataset with respect to the optimality criterion, a certain minimum number of objects is required. Referring to the above mentioned example (e.g. Fig. 3), this would mean that some incoming data of the computer network needs to be gathered.

Each object has an individual weight α_1, which is bounded by a parameter C. For the optimal representation the sum of the α_1 should be one. Given a very small set of objects, the

optimality criteria cannot be fullfilled.

Consider a simple example, where a minimum number of seven objects is required (see Fig. 4A to 4C). When the first six objects, plotted by stars in figure Fig. 4A are given maximal weight C, the optimality criterion cannot be fullfilled.

Suppose the window size is 100 examples and the expected outlier ratio is 7%. One can compute the value of $C = 1/7$. In order to bring the system in equilibrium, all the constraints must be satisfied; that is, all a_i should be $<= 1/7$ but their sum should be equal to one. It can be easily seen that these two constraints can only be satisfied after we have observed at least 7 points.

After adding a seventh object, indicated by the circle in Fig. 4B, its weight, and the weights of the other objects can be optimized (i.e. subjected to a minimisation routine to find an geometric representation. In this two-dimensional dataset a closed curve around the objects enclosing a minimal area).

The new object increases its weight α , while one of the other objects decreases its weight α to maintain the overall sum of the weights. These two objects are indicated by the 'x' marks in Fig. 4B.

In the final step of the optimization, the added object hits the upper weight bound. This is indicated in Fig. 4C by the change of the marker to a star.

The meaning of the curve in this figure, as well as in all subsequent figures, is the shape of the representation of normality. Although it may seem somewhat strange that there are no points inside the normality region, it should be noted, however, that the guarantees as to the upper bound on the number anomalies can be fulfilled only after at least n =

window_size points have been seen. Until then, although the feasible solution exists, the statistical features of this solution cannot be enforced.

In Fig. 5A to 5G the process of incorporating a new object to an existing classifier (i.e. an already existing geometric representation of normality 2200) is shown. As e.g. indicated in Fig. 5A there are some objects outside the closed curve 2200 which shows that those objects would be considered "anomalous".

Fig. 5A shows a scatterplot of twenty objects. On this dataset a classifier is trained (i.e. a minimisation as indicated above), and the geometric representation of normality 2200 as a decision boundary is plotted.

The three types of data objects are indicated:

- The dotted objects are the objects which are are classified as target objects (i.e. "normal"). These objects are said to belong to the 'rest' set, or set R. These objects have weight 0.

- The starred objects are objects rejected by the classifier (i.e. "anomalous"), and thus belong to the error set E. Their weights have the maximum value of C.

- Finally, the objects on the curve of the geometric representation of normality 2200 indicated by "x", are the support vectors (belonging to set S) which have a non-zero weigth, but are not bounded.

In Fig. 5B, a new object is added at position (2,0). This object is now added to the support set S, but the classifier is now out of equilibrium. In the following steps (see steps 2100, 2200, 2300 in Fig. 1) the weigths and the set memberships of the other objects are automatically adapted.

Until the system has reached the state of equilibrium, such geometric interpretation is not possible, which can be clearly seen starting from fig. 5b. We have added the new object to set S, in order to be able to change its weight; however, the curve cannot be immediately forces to go through the new object, and furthermore, at the beginning of the importation of the new object we do not know if it should pass through the new object. In fig. 5c and all subsequent figures the circle indicates the object that has changed its state. In the last figure, in which the new object has received its final state, one can see that the geometric representation is again consistent: the curve passes through the crosses and separates the stars (anomalies) from dots (normal points).

As can be seen from the above, the geometric representation of normality is updated sequentially which is essential for on-line (real time) applications. There are no prior assumptions about the classification. The classification (i.e. the membership to set) is developed automatically while the data is received.

In the next step (Fig. 5D), the same change is done by another object. After three more steps, the new equilibrium is obtained. Having this classifier, a new object can be processed now.

Figures 5D through 5G illustrate the progress of the algorithm and different possible state changes that the examples can undertake (see also by previous comment). In figure 5D the an object is removed from set S into set O. In figure 5E an object is added to set S from set E. In figure 5F an object is removed from set S into set E. Finally, in figure 5G a current object is assigned to set E and the equilibrium is reached.

In Appendix B, especially in section 2.4 a particular

advantageous formulation of the geometric representation of normality (2200), i.e. the quarter sphere is described. The asymmetry of the geometric representation of normality (2200) is well suited for data streams in intrusion problems.

For reasons of simplicity the inventive method and system is described in conncetion with a two-dimensional data set. Obviously the method and the system can be generalised to datasets with arbitrary dimensons. The curve would be a hypersurface enclosing a higher dimensional volume.

The invention is also applicable to monitoring of the measurements of physical parameters of operating mechanical devices, of the measurements of chemical processes and of the measurement of biological activity. In general the invention is specifically suited in situations in which continous data is received and no à priori classification or knwoledge about the source of the data is available.

Such an application is e.g. image analysis of medical samples where anomalous objects can be distinguished by a different colour or radiation pattern. Another possible medical application would be data streams representing electrical signals obtained from EEG or ECG apparatus. Here anomalous wave patterns can be automatically detected. Using EEG data the imminent occurence of an epileptic seizure might be detected.

Furthermore, data online colleced from mechanical or geophysical system can analyzed using the inventive method and system. Mechanical stress and resulting fractures can be discerned from the data. As soon as "anomalous" data (i.e. deviations from "normal" data) is received, this might indicate a noteworthy chance of conditions.

The inventive method and system could also be applied to pattern regognition in which the pattern is not known à

priori which is usually the case. The "anomalous" objects would be the ones not belonging to the pattern.

There is also a possible application of the inventive method and system in conncetion with financial data. It could be used to identify changes in tradeing data indicating unwanted risks. Credit card data could be also analyzed to identify risks or even fraud.

Appendix A describes a the general context of online SVM. Appendix B describes a special application using a quater-sphere method. Appendix C contains the description some extra Figure C2, C3, C5, C6, C7, C10, C11, C12. Fig. C2 gives general overview. Appendix D explains some of the formulae.

Claims

1. Method for automatic online detection and classification of anomalous objects in a data stream, especially comprising datasets and / or signals,

characterized in that

a) the detection of at least one incoming data stream (1000) containing normal and anomalous objects,

b) automatic construction (2100) of a geometric representation of normality (2200) the incoming objects of the data stream (1000) at a time t_1 subject to at least one predefined optimality condition, especially the construction of a hypersurface enclosing a finite number of normal objects,

c) online adaptation of the geometric representation of normality (2200) in respect to received at least one received object at a time $t_2 > t_1$, the adaptation being subject to at least one predefined optimality condition,

d) online determination of a normality classification (2300) for received objects at t_2 in respect to the geometric representation of normality (2200),

e) automatic classification of normal objects and anomalous objects based on the generated normality classification (2300) and generating a data set describing the anomalous data for further processing, especially a visual representation.

2. Method according to claim 1, characterised in that the geometric representation of normality (2200) is a parametric boundary hypersurface using the enclosure of the

minimal volume or minimal volume estimate among all possible surfaces as a optimality condition.

Method according to claim 2, characterised in that the parametric boundary hypersurface is the minimal volume among all possible surfaces consistent with a pre-defined condition, especially an expected fraction η of anomalous objects.

4. Method according to at least one preceding claim, characterised in that the anomalous objects are determined as the ones lying outside of the geometrical representation of normaliy (2200), especially the parametric boundary hypersurface enclosing the normal objects.

5. Method according to at least one preceding claim, characterized in that dynamic adaptation of the geometric representation of normality (2200) comprises an automatic adjustment of parameters x_i of the geometric representation of normality (2200) to incorporate at least one new object while maintaining the optimality of the geometric representation of normality (2200).

6. Method according to at least one preceding claim, characterized in that the dynamic adaptation of the geometric representation of normality (2200) comprises an automatic adjustment of parameters x_i of the geometric representation of normality (2200) to remove the least-relevant object while maintaining the optimality of the geometric representation of normality (2200).

7. Method according to at least one preceding claim, characterized in that the smallest volume geometric representation of normality (2200) is maintained from an instance t_i after which the construction of the geometric representation of normality (2200) is feasible subject to the optimality condition.

8. Method according to at least one preceding claim, characterized in that the geometric representation of normality (2200) is generated with a Support Vector Machine method, generating a parametric vector x to describe the representation.

9. Method according to at least one preceding claim, characterised in that the temporal change of the geometrical representation of normality (2200), especially the temporal change change of a parameter vector x of the geometrical representation of normality (2200) is stored for the evaluation of temporal trend in the data stream (1000).

10. Method accortind to at least on one preceeding claim, characterised in that the geometric representation of normality (2200) is a sphere, in particular a quarter sphere.

11. Method according to at least one preceding claim, characterized in that incoming data stream (1000) comprises data packets in communication networks or representations thereof.

12. Method according to claim 10, characterized in that the data packets comprise data from the logging in process in at least one computer.

13. Method according to claim 10 or 11, characterized in that the determination of normality of the received data packets distinguishes normal incoming data stream form anomalous data, especially sniffing attacks and / or denial of service attacks, whereby the means for automatic determining the normal and anomalous data generates a warning message.

14. System for automatic online detection and classification

of anomalous objects in a data stream, especially comprising datasets and / or signals,

characterized by

a) a detection means for least one incoming data stream (1000) containing normal and anomalous objects,

b) an automatic online anomaly detection engine comprising

- an automatic construction means (2100) of a geometric representation of normality (2200) for the incoming objects of the data stream (1000) at a time t_1 subject to at least one predefined optimality condition, especially for the construction of a hypersurface enclosing a finite number of normal objects, with an automatic online adaptation means for the geometric representation of normality (2200) in respect to received at least one received object at a time $t_2 > t_1$, the adaptation being subject to at least one predefined optimality condition, and

- an automatic online determination means of a normality classification (2300) for received objects at t_2 in respect to the geometric representation of normality (2200).

c) an automatic classifcation means (4000) of normal objects and anomalous objects based on the generated normality classification (2300) and generating a data set describing the anomalous data for further processing, especially a visual representation.

15. A method for construction and update of the geometric representation (any of the above) in which the coordinate system in which the representation is constructed is fixed to some point in the data or in the feature space.

16. A method according to claim 15, in which the center of coordinate system coinsides with the center of mass of the data (in the original or in the feature space) (see formulas for the centering)

17. A method according to claim 15 or 16, in which the decision on normality or anomality of an object is decided upon its norm in the data-centered (or feature-space-centered) coordinate coordinate system (or, equivalently by the radius of the hypersphere centered at the center of the origin in the said coordinate system and encompassing the given object).

18. A method according to one of the claims 15 to 17 in which the update of the representation includes the update of the coordinate system.

19. A method according to one of the claims 15 to 18 in which the update of coordinate system includes the update of the center of coordinates.

20 A method according to one of the claims 15 to 19 in which importation of the new object includes as a part the update of the norms of all objects in the working set so as to bring them in the new coordinate system corresponding to the expanded working set ("norm expansion")..

21. A method according to one of the claims 15 to 20, in which removal of the object includes as a part the update of the norms of all objects in the working set so as to bring them in the new coordinate system corresponding to the contracted working set ("norm contraction")

Abstract

The invention is concerned with a method for automatic online detection and classification of anomalous objects in a data stream, especially comprising datasets and / or signals,

characterized in that

a) the detection of at least one incoming data stream (1000) containing normal and anomalous objects,

b) automatic construction (2100) of a geometric representation of normality (2200) the incoming objects of the data stream (1000) at a time t_1 subject to at least one predefined optimality condition, especially the construction of a hypersurface enclosing a finite number of normal objects,

c) online adaptation of the geometric representation of normality (2200) in respect to received at least one received object at a time $t_2 > t_1$, the adaptation being subject to at least one predefined optimality condition,

d) online determination of a normality classification (2300) for received objects at t_2 in respect to the geometric representation of normality (2200),

e) automatic classification of normal objects and anomalous objects based on the generated normality classification (2300) and generating a data set describing the anomalous data for further processing, especially a visual representation.

Fig 1

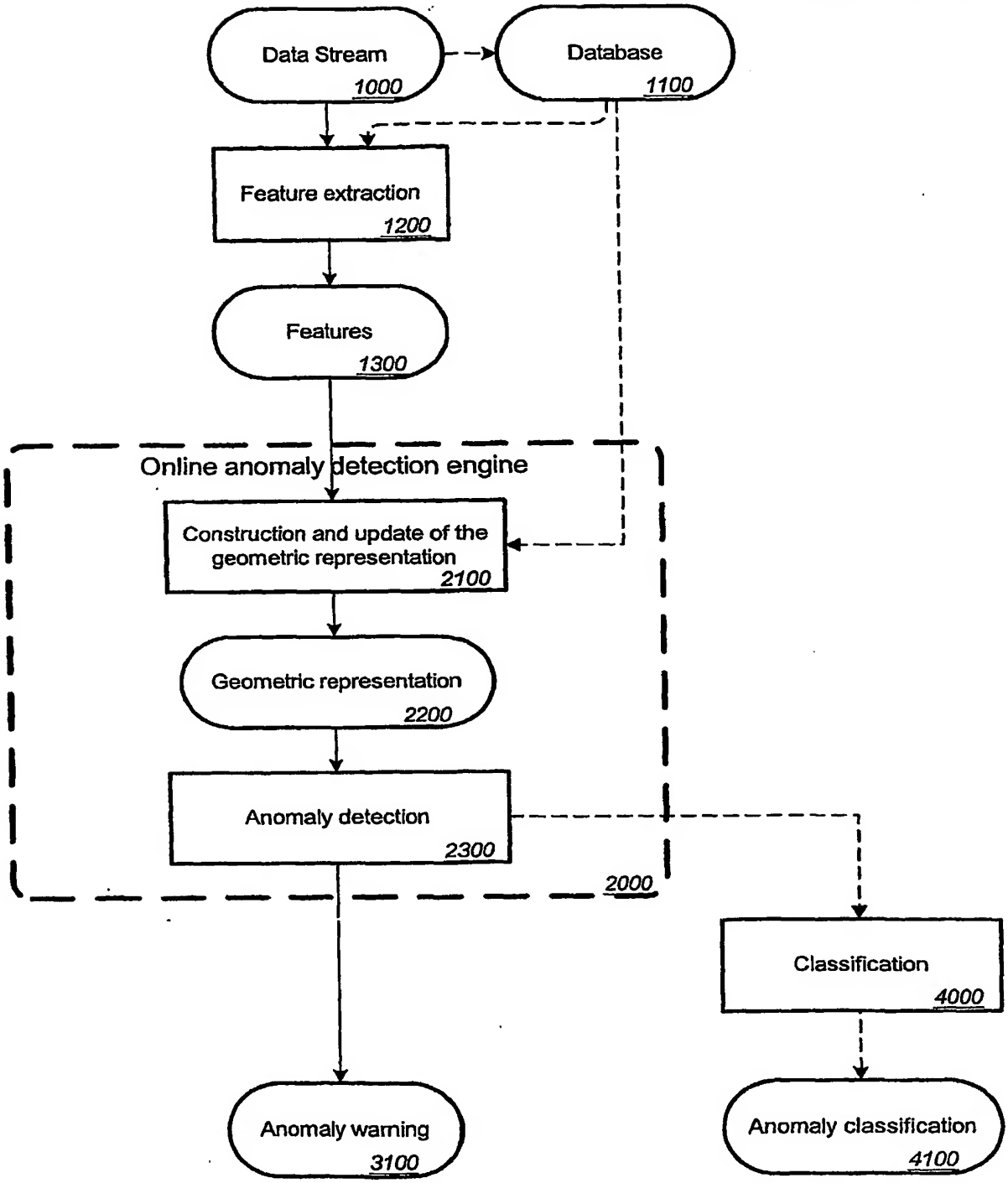
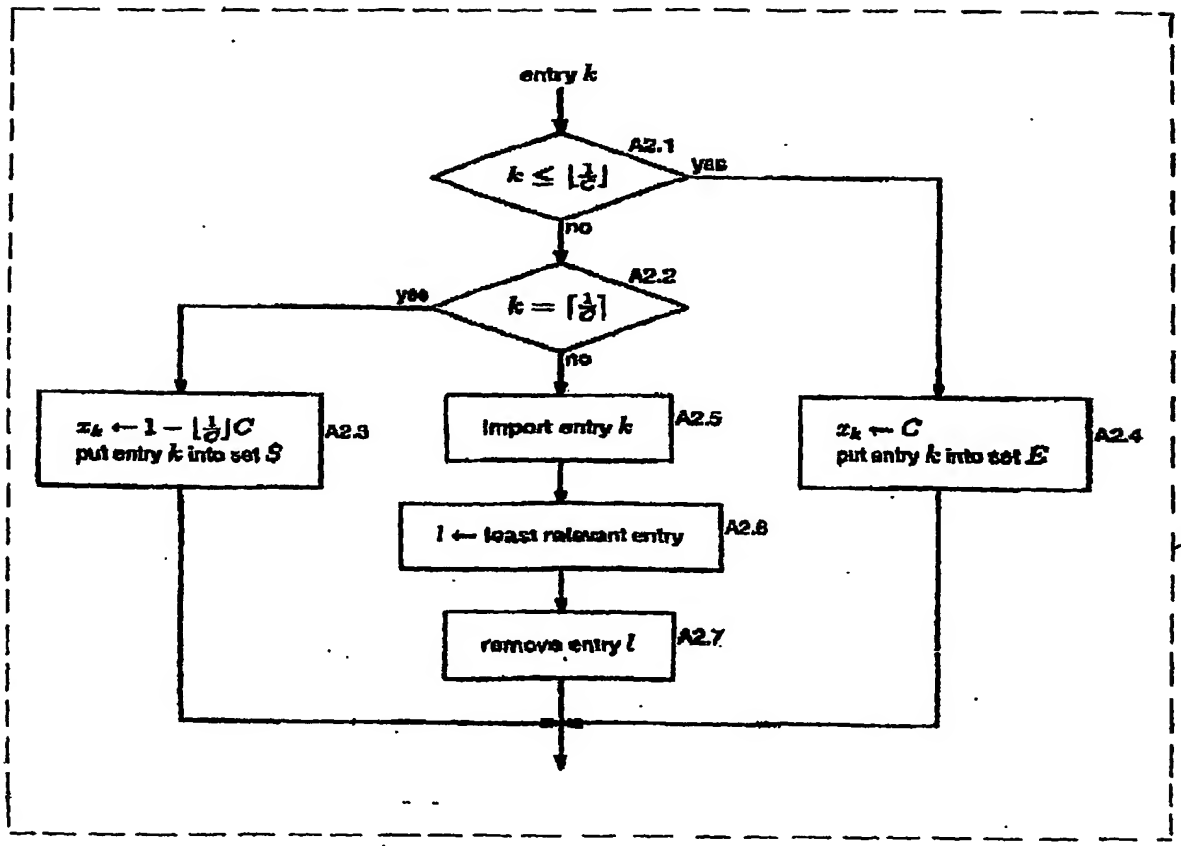


FIG 2


entry k
A2.1
$k \leq \lfloor \frac{1}{c} \rfloor$
yes
no
A2.2
$k = \lceil \frac{1}{c} \rceil$
yes
no
$x_k \leftarrow 1 - \lfloor \frac{1}{c} \rfloor C$
put entry k into set S
A2.3
Import entry k
A2.5
$x_k \leftarrow C$
put entry k into set E
A2.4
$l \leftarrow$ least relevant entry
A2.6
remove entry l
A2.7
A2

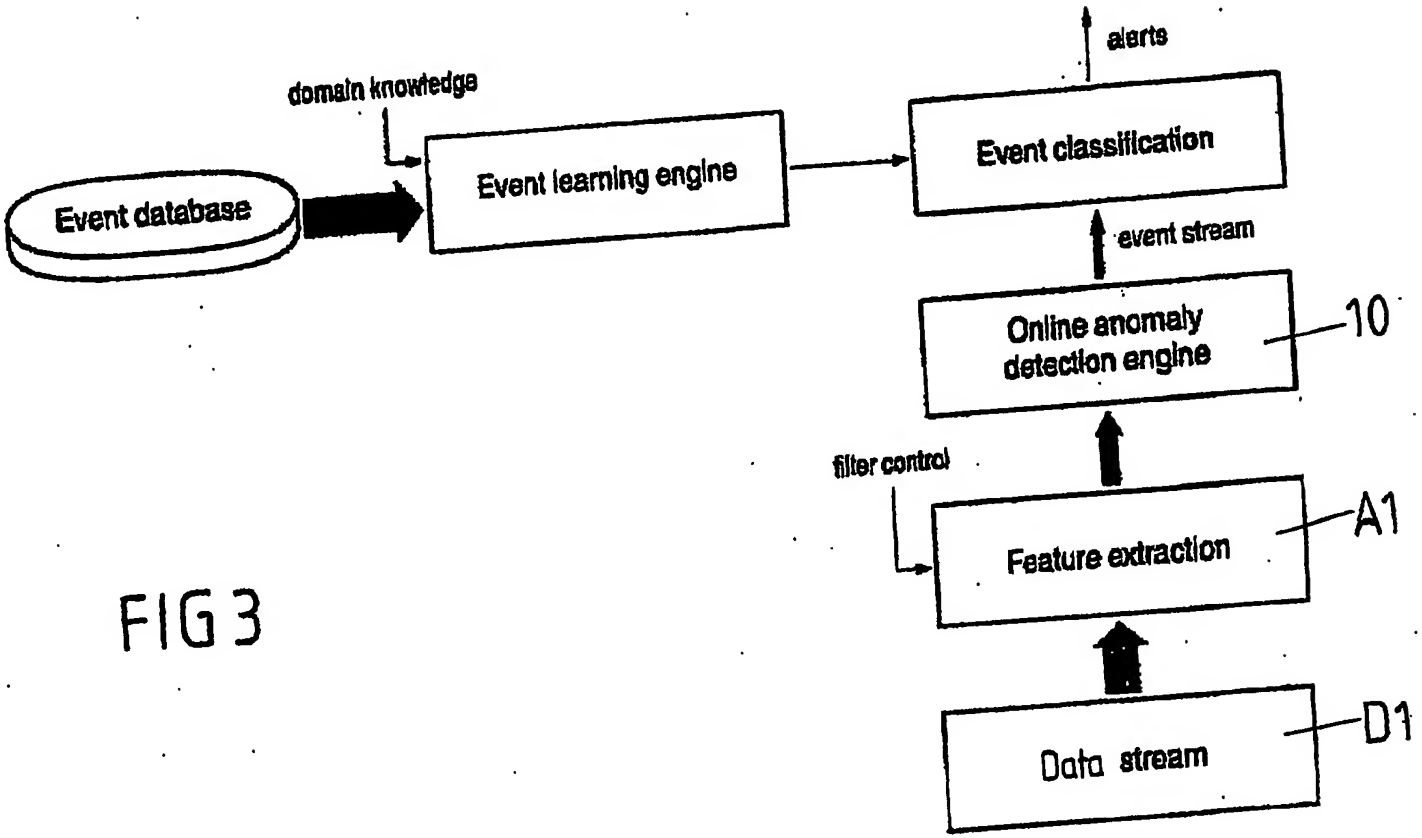

alerts
domain knowledge
Event learning engine
Event classification
Event database
event stream
Online anomaly detection engine
10
filter control
Feature extraction
A1
Data stream
D1

FIG 3

FIG 4A

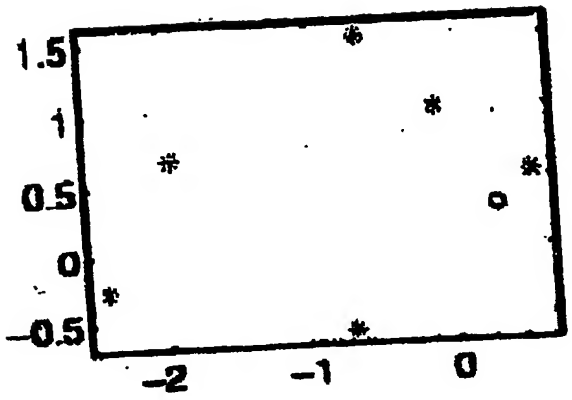


FIG 4B

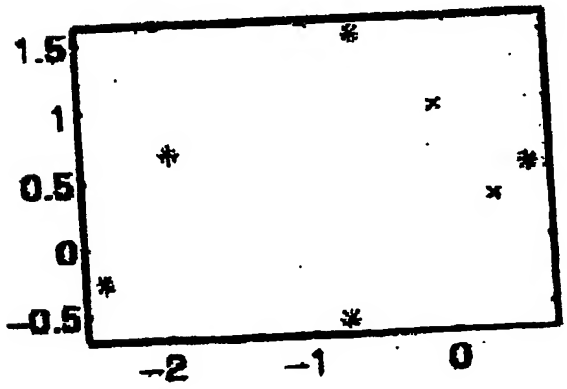


FIG 4C

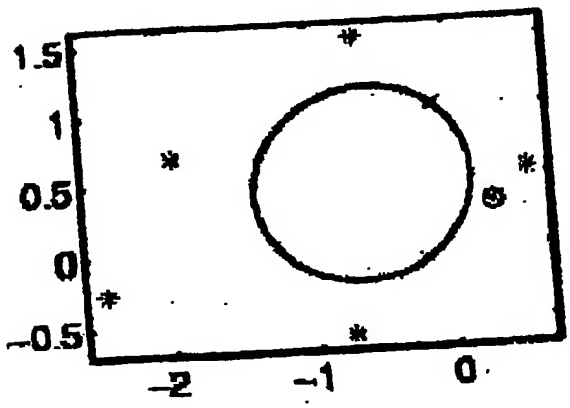


FIG 5A

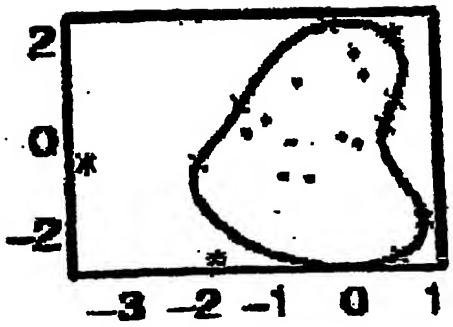


FIG 5B

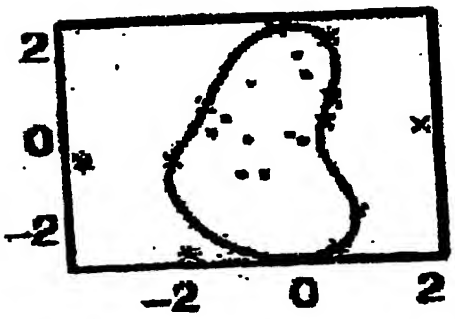


FIG 5C

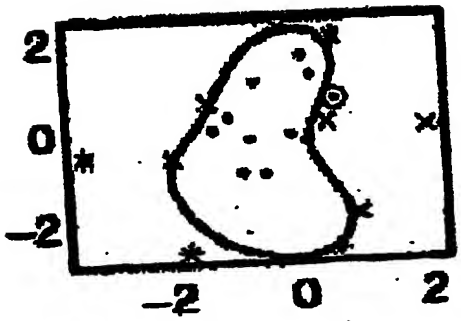


FIG 5D

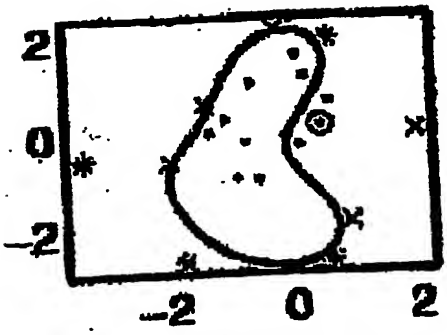


FIG 5E

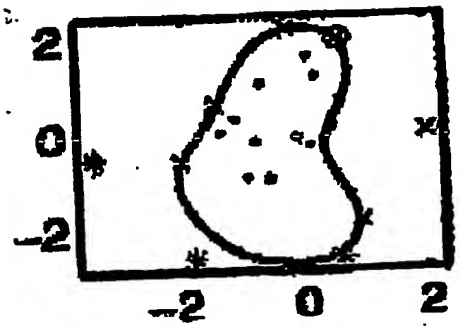


FIG 5F

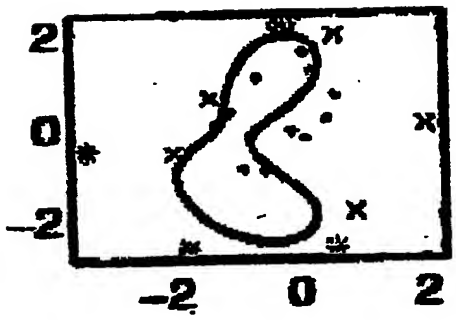


FIG 5G